Objektově orientované programování

Přednáška č. 2

Ing. Lukáš Slánský

Ústav elektrotechniky a informatiky, Univerzita Pardubice

12. 10. 2006

Obsah

Dědičnost

- Většinu tříd objektů v reálném světě lze uspořádat do hierarchie

Dědičnost

- Většinu tříd objektů v reálném světě lze uspořádat do hierarchie
  - Lednice, kávovar, mikrovlná trouba a další jsou

Dědičnost

- Většinu tříd objektů v reálném světě lze uspořádat do hierarchie
  - Lednice, kávovar, mikrovlná trouba a další jsou *kuchyňské spotřebiče*

Dědičnost

- Většinu tříd objektů v reálném světě lze uspořádat do hierarchie
  - Lednice, kávovar, mikrovlná trouba a další jsou *kuchyňské spotřebiče*
  - Kuchyňské spotřebiče, počítače, audiotechnika a další jsou

Dědičnost

- Většinu tříd objektů v reálném světě lze uspořádat do hierarchie
  - Lednice, kávovar, mikrovlná trouba a další jsou *kuchyňské spotřebiče*
  - Kuchyňské spotřebiče, počítače, audiotechnika a další jsou *elektrické spotřebiče*
  - ...

Dědičnost

- Většinu tříd objektů v reálném světě lze uspořádat do hierarchie
  - Lednice, kávovar, mikrovlná trouba a další jsou *kuchyňské spotřebiče*
  - Kuchyňské spotřebiče, počítače, audiotechnika a další jsou *elektrické spotřebiče*
  - ...
  - Dobře strukturovaný je živý svět, kde jsou jednotlivé třídy živočichů (druhy) zařazovány do říší, kmenů, řádů, čeledí, rodů, druhů, ...

Dědičnost

- Většinu tříd objektů v reálném světě lze uspořádat do hierarchie
  - Lednice, kávovar, mikrovlná trouba a další jsou *kuchyňské spotřebiče*
  - Kuchyňské spotřebiče, počítače, audiotechnika a další jsou *elektrické spotřebiče*
  - ...
  - Dobře strukturovaný je živý svět, kde jsou jednotlivé třídy živočichů (druhy) zařazovány do říší, kmenů, řádů, čeledí, rodů, druhů, ...
- Jedná se o hierarchii se vztahy typu ISA

Dědičnost

- Většinu tříd objektů v reálném světě lze uspořádat do hierarchie
  - Lednice, kávovar, mikrovlná trouba a další jsou *kuchyňské spotřebiče*
  - Kuchyňské spotřebiče, počítače, audiotechnika a další jsou *elektrické spotřebiče*
  - ...
  - Dobře strukturovaný je živý svět, kde jsou jednotlivé třídy živočichů (druhy) zařazovány do říší, kmenů, řádů, čeledí, rodů, druhů, ...
- Jedná se o hierarchii se vztahy typu ISA (is a = je)

Dědičnost

- Většinu tříd objektů v reálném světě lze uspořádat do hierarchie
  - Lednice, kávovar, mikrovlná trouba a další jsou *kuchyňské spotřebiče*
  - Kuchyňské spotřebiče, počítače, audiotechnika a další jsou *elektrické spotřebiče*
  - ...
  - Dobře strukturovaný je živý svět, kde jsou jednotlivé třídy živočichů (druhy) zařazovány do říší, kmenů, řádů, čeledí, rodů, druhů, ...
- Jedná se o hierarchii se vztahy typu ISA (is a = je)
- Hierarchii lze zaznamenat pomocí grafu tříd

ISA vztah

- Jedná se o hierarchii se vztahy typu ISA (is a = je)

ISA vztah

- Jedná se o hierarchii se vztahy typu ISA (is a = je)
  - Savec (zool. třída) je strunatec (zool. kmen), pták je strunatec, plaz je strunatec, ...

ISA vztah

- Jedná se o hierarchii se vztahy typu ISA (is a = je)
  - Savec (zool. třída) je strunatec (zool. kmen), pták je strunatec, plaz je strunatec, ...
  - Vztah ISA není *symetrický*

ISA vztah

- Jedná se o hierarchii se vztahy typu ISA (is a = je)
  - Savec (zool. třída) je strunatec (zool. kmen), pták je strunatec, plaz je strunatec, ...
  - Vztah ISA není *symetrický* – Tj. ne každý strunatec je savec (pták, plaz, ...)

ISA vztah

- Jedná se o hierarchii se vztahy typu ISA (is a = je)
  - Savec (zool. třída) je strunatec (zool. kmen), pták je strunatec, plaz je strunatec, ...
  - Vztah ISA není *symetrický* – Tj. ne každý strunatec je savec (pták, plaz, ...); je dokonce *antisymetrický*

ISA vztah

- Jedná se o hierarchii se vztahy typu ISA (is a = je)
  - Savec (zool. třída) je strunatec (zool. kmen), pták je strunatec, plaz je strunatec, ...
  - Vztah ISA není *symetrický* – Tj. ne každý strunatec je savec (pták, plaz, ...); je dokonce *antisymetrický*
  - Vztah ISA je *tranzitivní*

ISA vztah

- Jedná se o hierarchii se vztahy typu ISA (is a = je)
  - Savec (zool. třída) je strunatec (zool. kmen), pták je strunatec, plaz je strunatec, ...
  - Vztah ISA není *symetrický* – Tj. ne každý strunatec je savec (pták, plaz, ...); je dokonce *antisymetrický*
  - Vztah ISA je *tranzitivní* – Tj. stunatec je živočich (zool. říše), tedy i savci jsou živočichové

ISA vztah

- Jedná se o hierarchii se vztahy typu ISA (is a = je)
  - Savec (zool. třída) je strunatec (zool. kmen), pták je strunatec, plaz je strunatec, ...
  - Vztah ISA není *symetrický* – Tj. ne každý strunatec je savec (pták, plaz, ...); je dokonce *antisymetrický*
  - Vztah ISA je *tranzitivní* – Tj. stunatec je živočich (zool. říše), tedy i savci jsou živočichové
  - Vztah ISA je *reflexivní*

ISA vztah

- Jedná se o hierarchii se vztahy typu ISA (is a = je)
  - Savec (zool. třída) je strunatec (zool. kmen), pták je strunatec, plaz je strunatec, ...
  - Vztah ISA není *symetrický* – Tj. ne každý strunatec je savec (pták, plaz, ...); je dokonce *antisymetrický*
  - Vztah ISA je *tranzitivní* – Tj. stunatec je živočich (zool. říše), tedy i savci jsou živočichové
  - Vztah ISA je *reflexivní* – Tj. každý savec je savec

ISA vztah

- Jedná se o hierarchii se vztahy typu ISA (is a = je)
  - Savec (zool. třída) je strunatec (zool. kmen), pták je strunatec, plaz je strunatec, ...
  - Vztah ISA není *symetrický* – Tj. ne každý strunatec je savec (pták, plaz, ...); je dokonce *antisymetrický*
  - Vztah ISA je *tranzitivní* – Tj. stunatec je živočich (zool. říše), tedy i savci jsou živočichové
  - Vztah ISA je *reflexivní* – Tj. každý savec je savec
  - Z matematického pohledu je ISA vztah relací s vlastnostmi

ISA vztah

- Jedná se o hierarchii se vztahy typu ISA (is a = je)
  - Savec (zool. třída) je strunatec (zool. kmen), pták je strunatec, plaz je strunatec, ...
  - Vztah ISA není *symetrický* – Tj. ne každý strunatec je savec (pták, plaz, ...); je dokonce *antisymetrický*
  - Vztah ISA je *tranzitivní* – Tj. stunatec je živočich (zool. říše), tedy i savci jsou živočichové
  - Vztah ISA je *reflexivní* – Tj. každý savec je savec
  - Z matematického pohledu je ISA vztah relací s vlastnostmi částečného uspořádání

Vlastnosti tříd v hierarchii

Jaký je vztah mezi třídami "pes" a "savec"?

Vlastnosti tříd v hierarchii

Jaký je vztah mezi třídami "pes" a "savec"?

- Pes má všechny atributy savce

Vlastnosti tříd v hierarchii

Jaký je vztah mezi třídami "pes" a "savec"?

- Pes má všechny atributy savce
  - Některé další atributy může mít pes navíc

Vlastnosti tříd v hierarchii

Jaký je vztah mezi třídami "pes" a "savec"?

- Pes má všechny atributy savce
  - Některé další atributy může mít pes navíc
- Pes má všechny metody savce

Vlastnosti tříd v hierarchii

Jaký je vztah mezi třídami "pes" a "savec"?

- Pes má všechny atributy savce
  - Některé další atributy může mít pes navíc
- Pes má všechny metody savce
  - Některé další metody může mít pes navíc

Vlastnosti tříd v hierarchii

Jaký je vztah mezi třídami "pes" a "savec"?

- Pes má všechny atributy savce
  - Některé další atributy může mít pes navíc
- Pes má všechny metody savce
  - Některé další metody může mít pes navíc
  - Některé metody nemusí být přesně stejné u obecného savce i u obecného psa (podrobněji v dalších přednáškách)

Terminologie objektového programování

- Vztah mezi savcem a psem se nazývá *dědění* (angl. inheritance)

Terminologie objektového programování

- Vztah mezi savcem a psem se nazývá *dědění* (angl. inheritance)
  - Pes přebírá (dědí) od savce všechny jeho atributy a metody

Terminologie objektového programování

- Vztah mezi savcem a psem se nazývá *dědění* (angl. inheritance)
  - Pes přebírá (dědí) od savce všechny jeho atributy a metody
- Pes je ve vztahu k savci

Terminologie objektového programování

- Vztah mezi savcem a psem se nazývá *dědění* (angl. inheritance)
  - Pes přebírá (dědí) od savce všechny jeho atributy a metody
- Pes je ve vztahu k savci *potomek* (angl. descendant)

Terminologie objektového programování

- Vztah mezi savcem a psem se nazývá *dědění* (angl. inheritance)
  - Pes přebírá (dědí) od savce všechny jeho atributy a metody
- Pes je ve vztahu k savci *potomek* (angl. descendant)
- Savec je ve vztaku k psovi

Terminologie objektového programování

- Vztah mezi savcem a psem se nazývá *dědění* (angl. inheritance)
  - Pes přebírá (dědí) od savce všechny jeho atributy a metody
- Pes je ve vztahu k savci *potomek* (angl. descendant)
- Savec je ve vztaku k psovi *předek* (angl. ancestor)

Příklad použití dědičnosti v Pascalu

```
type CSavec=object
               private
                 Hmotnost:Real;
                 ...
               public
                 procedure krmSe;
                 procedure spi(JakDlouho:Real);
                 ...
             end;
```

Příklad použití dědičnosti v Pascalu

```
type CSavec=object
               private
                 Hmotnost:Real;
                 ...
               public
                 procedure krmSe;
                 procedure spi(JakDlouho:Real);
                 ...
             end;
     CPes=object
             private
               ...
             public
               procedure stekej;
               ...
           end;
```

Příklad použití dědičnosti v Pascalu

```
type CSavec=object
               private
                 Hmotnost:Real;
                 ...
               public
                 procedure krmSe;
                 procedure spi(JakDlouho:Real);
                 ...
             end;
     CPes=object(CSavec)
             private
               ...
             public
               procedure stekej;
               ...
           end;
```

Omezení dědění objektů

- Dědit lze pouze od jednoho předka

Omezení dědění objektů

- Dědit lze pouze od jednoho předka
  - Existují jazyky s vícenásobnou dědičností – např. C++

Omezení dědění objektů

- Dědit lze pouze od jednoho předka
  - Existují jazyky s vícenásobnou dědičností – např. C++
- Atributy ani metody nelze v potomcích zapomínat

Omezení dědění objektů

- Dědit lze pouze od jednoho předka
  - Existují jazyky s vícenásobnou dědičností – např. C++
- Atributy ani metody nelze v potomcích zapomínat
  - Lze měnit implementaci metod

Omezení dědění objektů

- Dědit lze pouze od jednoho předka
  - Existují jazyky s vícenásobnou dědičností – např. C++
- Atributy ani metody nelze v potomcích zapomínat
  - Lze měnit implementaci metod – předefinováním nebo polymorfismem (podrobněji v dalších přednáškách)

Omezení dědění objektů

- Dědit lze pouze od jednoho předka
  - Existují jazyky s vícenásobnou dědičností – např. C++
- Atributy ani metody nelze v potomcích zapomínat
  - Lze měnit implementaci metod – předefinováním nebo polymorfismem (podrobněji v dalších přednáškách)
  - Některé dobré objektové jazyky (Java, C++, ne Pascal) mohou rozlišovat mezi metodami (i funkcemi) podle počtu a typu parametrů a příp. i návratové hodnoty

Kdy používat dědění?

Kdy používat dědění?

- Existuje-li hierarchický ISA vztah mezi třídami

Kdy používat dědění?

- Existuje-li hierarchický ISA vztah mezi třídami

Správné použití:

Kdy používat dědění?

- Existuje-li hierarchický ISA vztah mezi třídami

Správné použití:

- Třída *člověk* je předkem pro třídy *zaměstnanec* i *zákazník*

Kdy používat dědění?

- Existuje-li hierarchický ISA vztah mezi třídami

Správné použití:

- Třída *člověk* je předkem pro třídy *zaměstnanec* i *zákazník*
  - Protože zaměstnanec i zákazník jsou lidé...

Kdy používat dědění?

- Existuje-li hierarchický ISA vztah mezi třídami

Správné použití:

- Třída *člověk* je předkem pro třídy *zaměstnanec* i *zákazník*
  - Protože zaměstnanec i zákazník jsou lidé...

Špatné použití:

Kdy používat dědění?

- Existuje-li hierarchický ISA vztah mezi třídami

Správné použití:

- Třída *člověk* je předkem pro třídy *zaměstnanec* i *zákazník*
  - Protože zaměstnanec i zákazník jsou lidé...

Špatné použití:

- Třída *bod* je předkem pro třídy *kružnice* nebo *úsečka*

Kdy používat dědění?

- Existuje-li hierarchický ISA vztah mezi třídami

Správné použití:

- Třída *člověk* je předkem pro třídy *zaměstnanec* i *zákazník*
  - Protože zaměstnanec i zákazník jsou lidé...

Špatné použití:

- Třída *bod* je předkem pro třídy *kružnice* nebo *úsečka*
  - Kružnice (nebo úsečka) *není* bod

Kdy používat dědění?

- Existuje-li hierarchický ISA vztah mezi třídami

Správné použití:

- Třída *člověk* je předkem pro třídy *zaměstnanec* i *zákazník*
  - Protože zaměstnanec i zákazník jsou lidé...

Špatné použití:

- Třída *bod* je předkem pro třídy *kružnice* nebo *úsečka*
  - Kružnice (nebo úsečka) *není* bod
  - Kružnice (i úsečka) obsahuje bod

Kdy používat dědění?

- Existuje-li hierarchický ISA vztah mezi třídami

Správné použití:

- Třída *člověk* je předkem pro třídy *zaměstnanec* i *zákazník*
  - Protože zaměstnanec i zákazník jsou lidé...

Špatné použití:

- Třída *bod* je předkem pro třídy *kružnice* nebo *úsečka*
  - Kružnice (nebo úsečka) *není* bod
  - Kružnice (i úsečka) obsahuje bod – je potřeba mít mezi třídami bod a kružnice jiný typ vztahu, ne ISA